2.1 - Overall Resource Loading Data

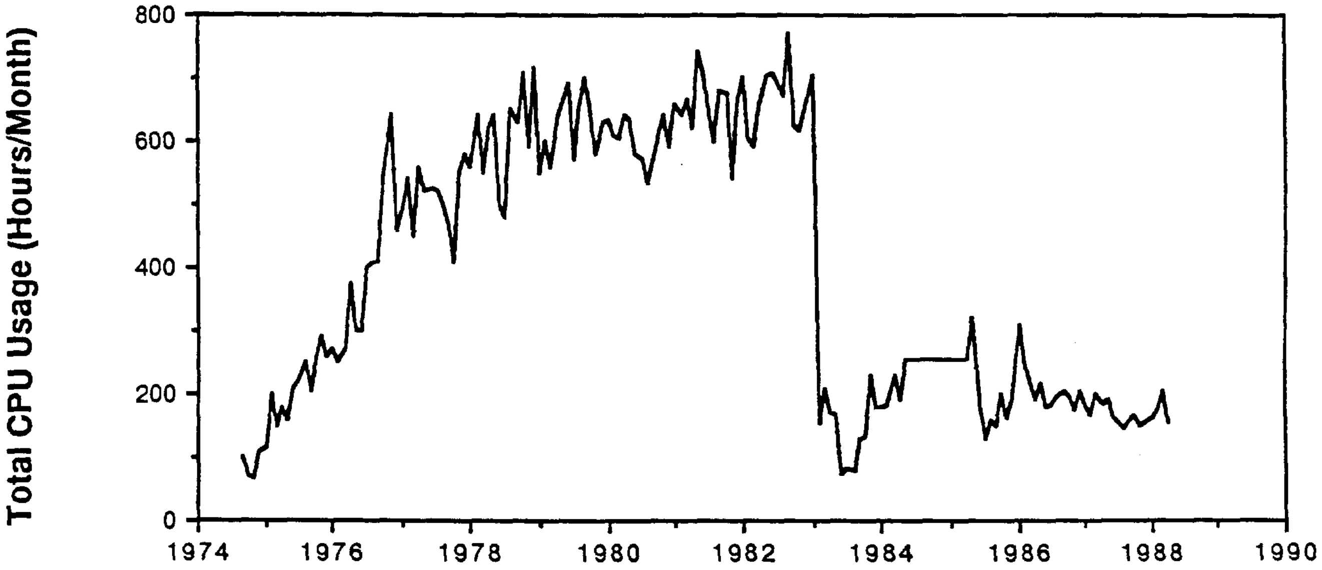

Figure 10: Total CPU Hours Consumed by Month

2.2 - Relative System Loading by Community

The SUMEX resource is divided, for administrative purposes, into three major communities: core ONCOCIN research, core AI research, and user projects based at the Stanford Medical School (*Stanford Projects*); user projects based outside of Stanford (*National AIM Projects*); and core system development efforts (*System Staff*). The initial resource management plan approved by the BRTP at the start of SUMEX specified that available system CPU capacity and file space resources were to be divided nominally between these communities in a 40:40:20 ratio. The "available" resources are those remaining after various monitor and community-wide functions (e.g., job scheduling, system overhead, network service, file space for subsystems, documentation, etc.) are accounted for.

The monthly usage of CPU resources and terminal connect time for each of these three communities is shown in the plots in Figure 11 and Figure 12. Many of the national user projects have already moved to their own machines for intensive research computing and now use SUMEX mainly for communications and information access. Hence, one might expect the proportion of CPU and file space use by Stanford projects, as compared to non-Stanford groups, to continue to grow correspondingly, as has been the case in the past. However, this past year there has been a dramatic increase in the national use of SUMEX-AIM for remote communications and information access. Much of this has been through the "anonymous" file transfer mechanism for which we cannot identify the user by name. We will attempt to record more information about such information access connections in future years.

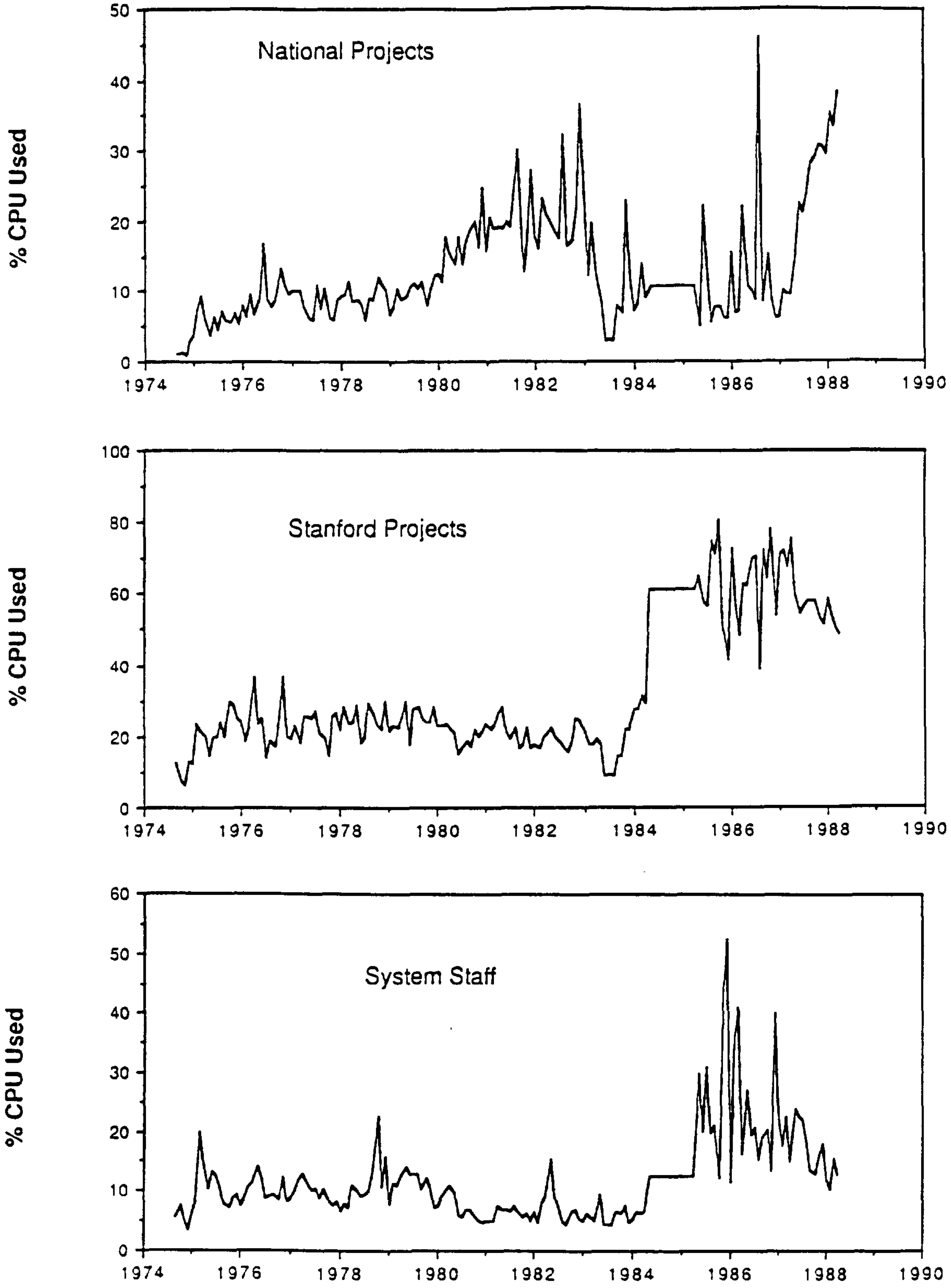

Figure 11: Monthly CPU Usage by Community

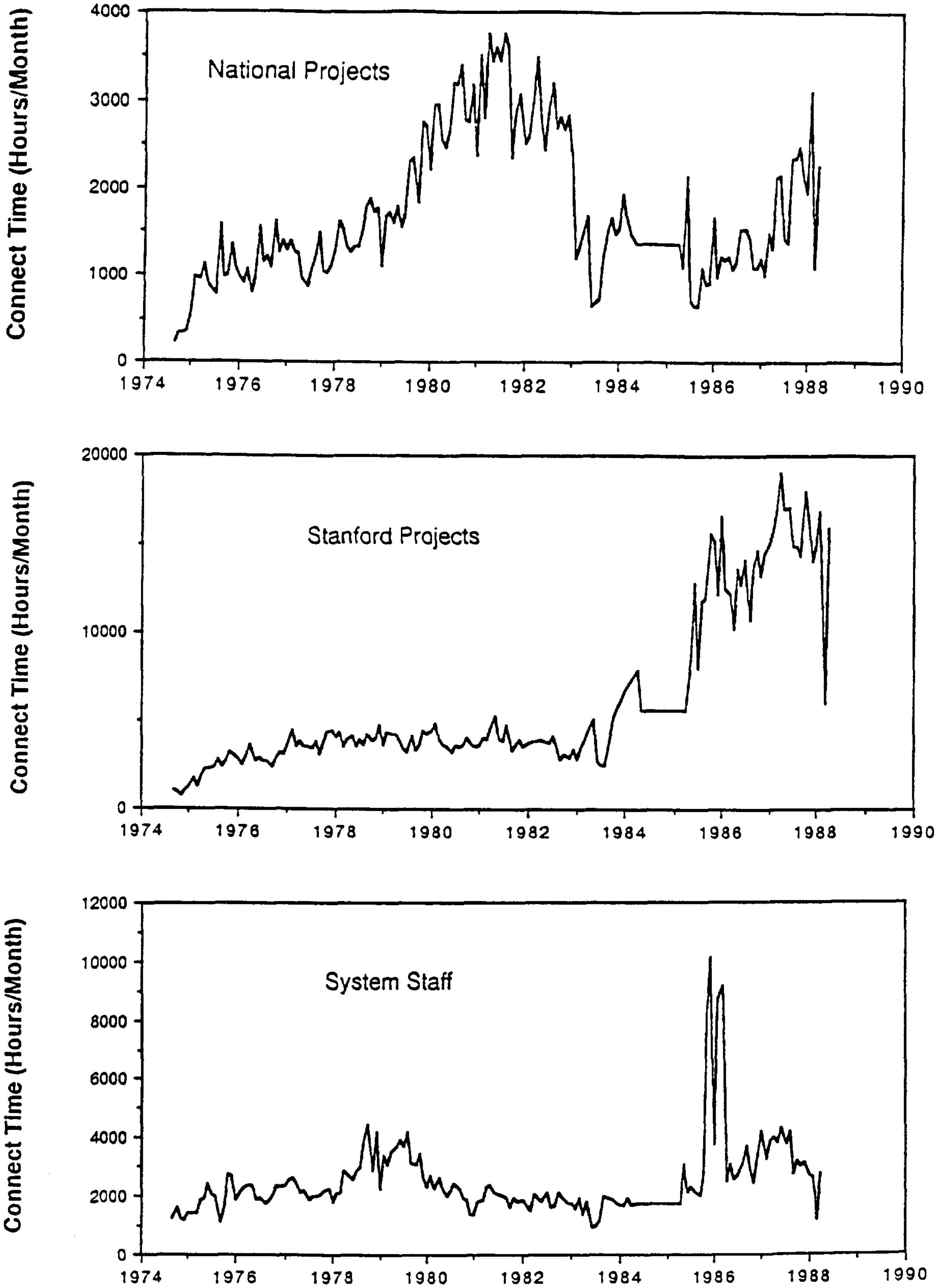

Figure 12: Monthly Terminal Connect Time by Community

2.3 - Individual Project and Community Usage

The following histogram and table show cumulative resource usage by collaborative project and community during the past grant year. The histogram displays the project distribution of the total CPU time consumed between May 1, 1987 and April 30, 1988, on the SUMEX-AIM DECsystem 2060 system. Data include total CPU consumption by project (Hours), total terminal connect time by project (Hours), and average file space in use by project (Pages, 1 page = 512 computer words). These data were accumulated for each project for the months between May 1986 and April 1987.

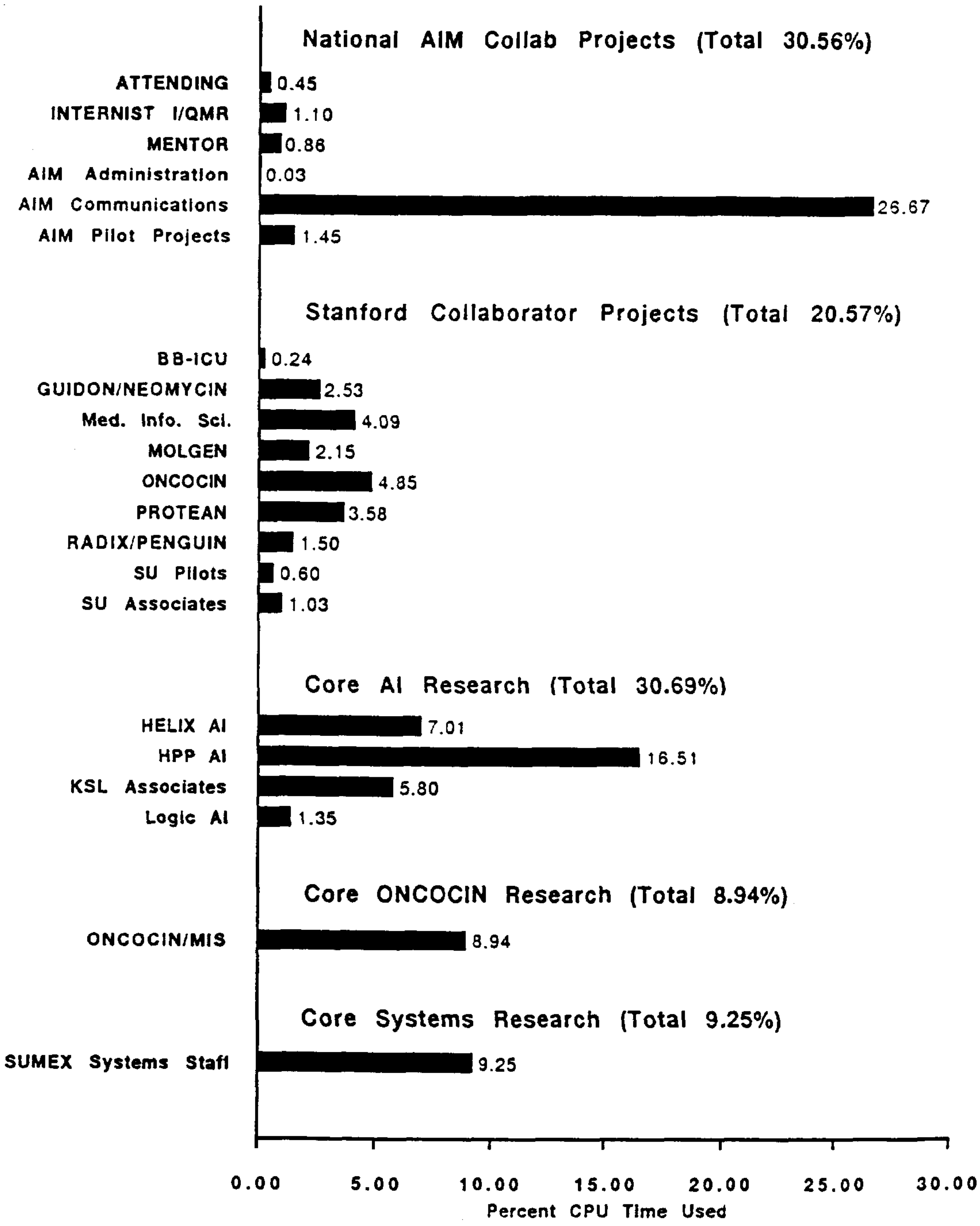

Figure 13: Cumulative CPU Usage Histogram by Project and Community

Resource Use by Individual Project - 5/87 through 4/88

	CPU	Connect	File Space
National AIM Collaborator Community	(Hours)	(Hours)	(Pages)
1) ATTENDING "A Critiquing Approach to Expert Computer Advice" Perry L. Miller, M.D., Ph.D. Yale University School of Medicine	5.08	217.81	650
2) INTERNIST-QMR Project "Clinical Decision Systems Research Resource" Jack D. Myers, M.D. Randolph A. Miller, M.D. University of Pittsburgh	12.37	259.59	4693
3) MENTOR Project "Medical Evaluation of Therapeutic Orders" Stuart M. Speedie, Ph.D. University of Maryland Terrence F. Blaschke, M.D. Stanford University	9.62	4830.97	2000
4) AIM Pilot Projects			
PathFinder (Nathwani and Fagan)	7.04	1002.52	1560
Dynamic Systems (Widman)	9.19	224.45	1433
Radiation Therapy (Kalet)	0.02	1.42	4
5) AIM Communications			
AIM Mail-Only Users	6.64	970.79	3567
AAAI Management	5.50	1930.65	929
BIONET	2.32	168.76	680
MCS Collaborators	7.25	1796.62	1110
MOLGEN Collaborators	2.99	310.63	883
Anonymous File/ Information Access	273.22	13593.75	233579
Other	0.62	32.77	839
6) AIM Administration	0.32	44.53	2009
	---------	----------	-------
Community Totals	342.18	25385.27	253936

Figure 14: Table of Resource Use by Project

	CPU	Connect	File Space
Stanford Collaborator Community	(Hours)	(Hours)	(Pages)
1) BBICU Project Lawrence M. Fagan, M.D., Ph.D. Department of Medicine Barbara Hayes-Roth, Ph.D. Computer Science Department	2.65	1512.22	237
2) GUIDON-NEOMYCIN Project William J. Clancey, Ph.D. Bruce G. Buchanan, Ph.D. Dept. Computer Science	28.37	6359.48	1862
3) Medical Information Sciences Edward H. Shortliffe, M.D., Ph.D. Lawrence M. Fagan, M.D., Ph.D. Department of Medicine	45.76	11821.74	4128
4) MOLGEN Project "Applications of Artificial Intelligence to Molecular Biology: Research in Theory Formation, Testing and Modification" Edward A. Feigenbaum, Ph.D. Peter Friedland, Ph.D. Charles Yanofsky, Ph.D. Depts. Computer Science/Biology	24.04	6523.52	7114
5) ONCOCIN Project "Knowledge Engineering for Medical Consultation" Edward H. Shortliffe, M.D., Ph.D. Lawrence M. Fagan, M.D., Ph.D. Department of Medicine	54.31	11759.34	7940
6) PROTEAN Project Oleg Jardetzky School of Medicine Bruce Buchanan Computer Science Department	40.04	8795.76	4097
7) RADIX-PENGUIN Project Gio C.M. Wiederhold, Ph.D. Depts. Computer Science/ Medicine	16.84	3150.80	9284

Figure 14: Table of Resource Use by Project, Continued

8) Stanford Pilot Projects REFEREE Project (Buchanan)	6.72	1380.48	452
9) Stanford Associates	11.55	8064.25	3265
	---------	----------	-------
Community Totals	230.29	59367.60	38380

Core AI Research	CPU (Hours)	Connect (Hours)	File Space (Pages)
1) ABLE Project Robert S. Engelmore, Ph.D. Computer Science Department Scott Clearwater, Ph.D. Los Alamos National Laboratory	11.85	5372.31	570
2) Advanced Architectures Edward A. Feigenbaum, Ph.D. Computer Science Department	105.23	32008.01	7284
3) Blackboard Architectures Barbara Hayes-Roth, Ph.D. Computer Science Department	53.33	10390.84	4636
4) DART Project Michael R. Genesereth, Ph.D. Computer Science Department	6.10	3220.27	0
5) Financial Resource Management Bruce G. Buchanan, Ph.D. Thomas C. Rindfleisch Computer Science Department	26.07	7449.24	2180
6) Intelligent Agents Project Michael R. Genesereth, Ph.D. Computer Science Department	6.32	507.92	0
7) Knowledge Engineering Studies Bruce G. Buchanan, Ph.D. Dianna Forsythe, Ph.D. Computer Science Department	5.66	1313.59	236
8) Machine Learning Studies Bruce G. Buchanan, Ph.D. Computer Science Department	34.23	9774.46	6134

Figure 14: Table of Resource Use by Project, Continued

9) MRS Project Michael R. Genesereth, Ph.D. Computer Science Department	2.75	1336.60	0
10) SOAR Project Paul R. Rosenbloom, Ph.D. Information Sciences Institute University of Southern California	12.57	6979.07	907
11) Software Design Project H. Penny Nii Computer Science Department	1.13	124.99	904
12) Very Large Knowledge Bases Edward A. Feigenbaum, Ph.D. Richard Keller, Ph.D. Computer Science Department	13.33	3153.20	1298
13) KSL Administration	55.15	13910.04	10090
14) KSL Associates	9.84	1645.64	1554
	---------	----------	-------
Community Totals	343.56	97186.17	35793

Core ONCOCIN Research	CPU (Hours)	Connect (Hours)	File Space (Pages)
1) Core ONCOCIN and Medical Information Sciences Edward H. Shortliffe, M.D., Ph.D. Lawrence M. Fagan, M.D., Ph.D. Department of Medicine	100.07	23581.08	12068
	---------	----------	-------
Community Totals	100.07	23581.08	12068

Core Systems Research	CPU (Hours)	Connect (Hours)	File Space (Pages)
1) SUMEX Staff R & D Thomas C. Rindfleisch Departments of Medicine and Computer Science	98.27	18547.71	10008

Figure 14: Table of Resource Use by Project, Continued

2) System Associates	5.24	252.56	1201
	---------	----------	-------
Community Totals	103.51	18800.29	11209
System Operations	CPU (Hours)	Connect (Hours)	File Space (Pages)
1) System Operations	650.01	80316.52	3437
2) SUMEX Staff Opns & Mgmnt	98.27	18547.71	10008
	---------	----------	-------
Community Totals	748.28	98864.23	13445
	=========	==========	=======
Resource Grand Totals	1867.89	323184.64	364831

Figure 14: Table of Resource Use by Project, Concluded

2.4 - Network Usage Statistics

The plots in Figures 15 and 16 show the monthly network terminal connect time for the public data networks and the INTERNET usage. The INTERNET is a broader term for what was previously referred to as ARPANET usage. Since many vendors now support the INTERNET protocols (TCP/IP) in addition to the ARPANET, which converted to TCP/IP in January of 1983, it is no longer possible to distinguish between ARPANET usage and Internet usage on our 2060 system. Similarly, after we switched to the Develcon gateway between the TELENET X.25 network and our TCP/IP Ethernet, we are not able to distinguish TELENET 2060 users from other Internet users. We are hoping to refine the accounting services available from the Develcon gateway so we will have a separate log of connection activity.

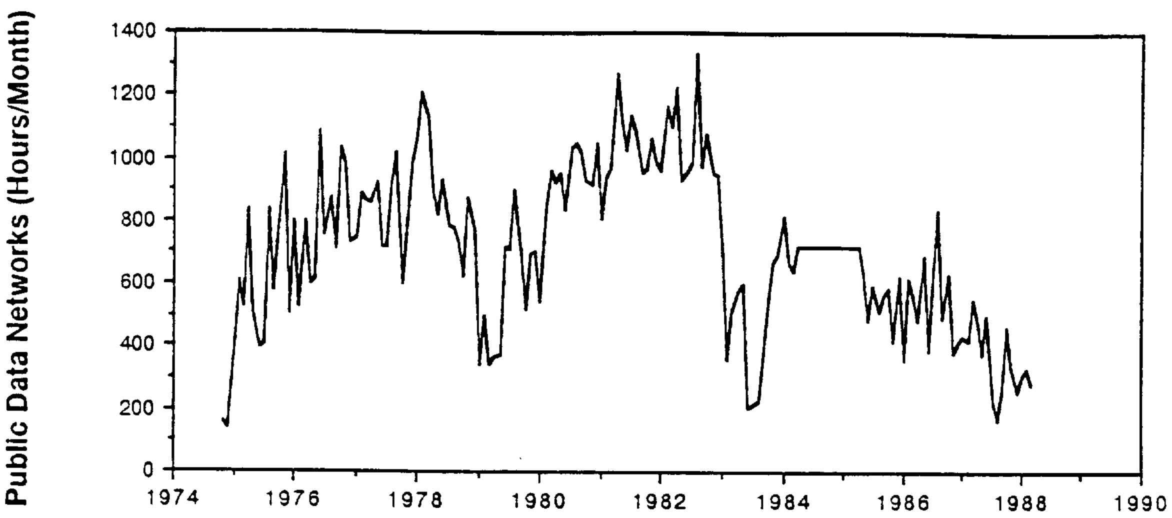

Figure 15: Public Data Network Terminal Connect Time

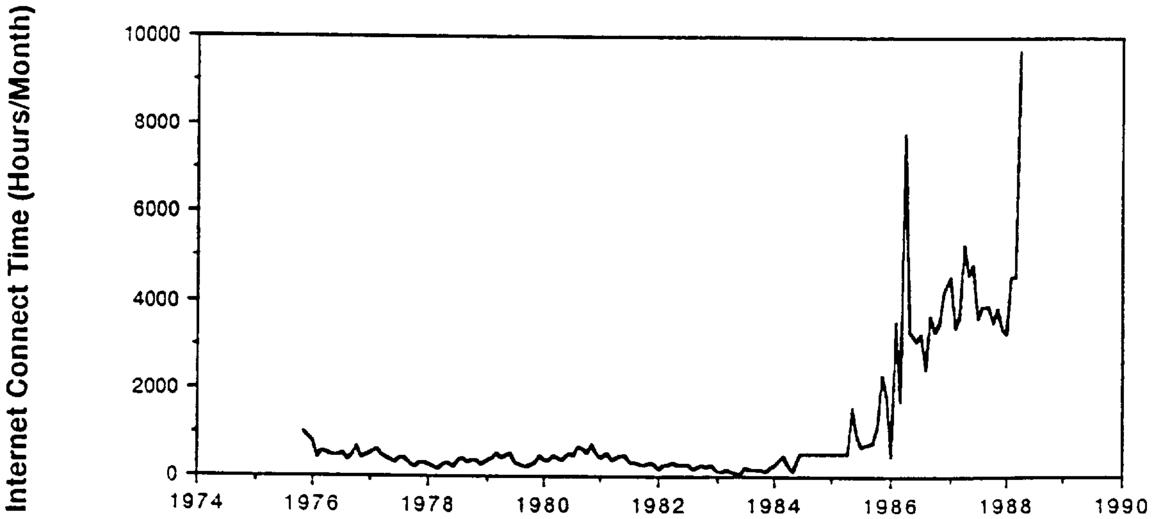

Figure 16: INTERNET Terminal Connect Time

2.5 - System Reliability

As in past years, the reliability of the 2060 system has been very high. The data below cover both hardware- and software-related system failures. The time listed under preventive maintenance (PM) downtime includes both the time required for hardware PM and the time required for installation of updated system software. The data cover the period of May 1, 1987 to April 30, 1988.

May 1987 - April 1988:

May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
14.2	3.2	5.0	0.1	24.0	10.9	2.6	4.0	5.4	2.4	35.2	2.2

Figure 17: 2060 Downtime Summary -- Hours per Month

May 1987 - April 1988:

Reporting period:	366 days, 0 hours, 12 min
Total Uptime:	361 days, 10 hours, 56 min
Uptime Percentage:	98.99%
PM Downtime:	0 days, 20 hours, 9 min
Actual Downtime:	3 days, 17 hours, 6 min
Total Downtime:	4 days, 13 hours, 16 min
MTBF:[1]	5 days, 15 hours, 32 min

Figure 18: Overall 2060 Reliability Summary

[1] Mean Time Between Failures

III.B. Highlights

In this section we describe several research highlights from the past year's activities. These include notes on existing projects that have passed important milestones, new pilot projects that have shown progress in their initial stages, and other core research and special activities that reflect the progress, impact, and influence the SUMEX-AIM resource has had in the scientific and educational communities.

III.B.1. PROTEGE -- Developing Knowledge Acquisition Tools for Clinical Trial Advice Systems

Knowledge acquisition, the process whereby computer scientists (*knowledge engineers*) interview experts in a given application area and attempt to encode the experts' specialized knowledge in a computer program, is widely recognized as a principal obstacle in the development of knowledge-based systems. To ease these difficulties, workers in medical artificial intelligence have experimented with a number of computer-based tools designed to facilitate the construction of clinical advice systems. One such tool is OPAL, a program that allows physicians to enter descriptions of oncology treatment plans directly into the knowledge base of ONCOCIN, an expert system that offers advice concerning protocol-directed cancer therapy. Physicians who use OPAL do not have to understand the production rules and other data structures that are used internally by ONCOCIN to represent cancer-therapy knowledge. Rather, oncologists enter knowledge into OPAL by drawing flowcharts and by filling in the "blanks" of graphical forms that anticipate the concepts required to define cancer therapy protocols. OPAL automatically converts the physicians' specifications into the knowledge representations that ONCOCIN uses to generate its treatment advice.

In 1986, system builders entered 36 oncology protocols into the ONCOCIN knowledge base using OPAL. Encoding these protocols via traditional knowledge engineering techniques might well have taken several person-years. Although OPAL clearly streamlines knowledge entry for the ONCOCIN system, OPAL itself required about 3.5 person-years of software development before it was ready for routine use. Furthermore, because OPAL takes advantage of a detailed model of therapy planning in oncology, the program is very much domain-dependent. OPAL is of no use, for example, to endocrinologists who desire to create knowledge bases for therapy planning for thyroid disease or to cardiologists interested in treating heart failure.

PROTEGE is a system that allows expert system developers to create knowledge acquisition tools that are much like OPAL (i.e., have convenient and powerful user interfaces), but that are custom tailored for new application areas. PROTEGE permits knowledge engineers to define models of the *kinds of clinical trials* that occur in various areas of medicine. It then uses these models to produce domain-specific knowledge acquisition tools that allow physicians to define new protocols by filling in graphical forms and by drawing flowchart diagrams. To date, PROTEGE has been used to create two such tools: 1) p-OPAL -- a program that incorporates most of the functionality of OPAL and thus acquires knowledge concerning clinical trials in oncology and 2) HTN -- a knowledge acquisition tool for hypertension drug studies. Producing each of these knowledge acquisition tools required a knowledge engineer using PROTEGE to enter *models* for the relevant classes of clinical trials (oncology and hypertension, respectively), defining those clinical trial models in terms of a general model of treatment planning built into the PROTEGE system itself. The PROTEGE user fills in the blanks of various forms to define models for given types of clinical trial applications, much as the user of OPAL fills out graphical forms to define individual cancer protocols. The forms in PROTEGE directly reflect the general model of treatment planning. Once the clinical trial models had been specified, PROTEGE generates the corresponding knowledge acquisition tools (computer programs) automatically.

III.B.2. A Speech Interface to ONCOCIN

Motivated by a recurrent request from collaborating clinicians to augment the "classical" keyboard interface to expert systems with a speech-based interface and by recent technological advances in continuous-speech systems, we began a project to explore the integration of speech input with the ONCOCIN cancer therapy advisor system. The project uses a commercial continuous-speech system loaned to us by Speech Systems, Inc. (SSI) of Tarzana, California. The speech recognizer consists of a custom microelectronic processor and a suite of special speech decoding software modules.

This experiment has taken advantage of our on-going work in distributed computing since the phonetic device, initial parsing software, and the ONCOCIN system all run on different pieces of hardware. Researchers have developed a prototype network connection and command interpreter between the speech module (running on a SUN workstation) and the Xerox 1186 computer that runs ONCOCIN and have designed a series of modifications to the ONCOCIN user interface has in turn been modified to accept verbal commands.

The prototype interface system permits users to navigate the graphical ONCOCIN interface and enter clinical data using speech. The system uses the location of the cursor on the screen to provide a context for choosing candidate grammars with which to attempt to recognize the user's utterance. The system dynamically adjusts the list of candidate recognition grammars based on the on-going dialog and it is now possible to carry on most of the ONCOCIN data acquisition steps using speech alone or speech plus pointing with the mouse. In addition, some input data elements (such as the neural toxicities) can be entered as textual descriptions and automatically encoded on the 1-4 point numerical scale used on oncology flowsheet forms.

We are also performing experiments to enhance the system's grammars with a wider range of phrases clinicians actually use when talking to a computer and to gain insights into clinicians' models of spoken interaction with advice systems. This will allow us to ground our interface design better in observed practice. In order to assess how physicians would speak to a computer in an ideal situation, we are simulating fully functional continuous-speech understanding with a hidden computer operator generating the output of ONCOCIN as if it had the ability to understand all spoken input. A video camera records both audio and visual clues. The physicians use ONCOCIN in the same manner as it is used in the clinic when they see patients, but with the added capability of (simulated) speech input. These experiments enable us to build up both a basic vocabulary for the real speech system as well as examine subtle linguistic issues to guide future directions.

III.B.3. SIMPLE/CARE -- Emulation of Parallel Computing Architectures

Many applications require knowledge-based systems that can cope with large amounts of data and produce responses in real-time. The current hardware and software architectures for knowledge-based systems cannot support such requirements. The most promising approach for achieving orders of magnitude improvement in the quantitative performance of knowledge-based systems is by exploiting concurrency on multiprocessor systems. Based on projections for integrated circuit technologies, it is clear that highly parallel multiprocessor computers, consisting of 100's to 1000's of processors and realizing a variety of concurrent architectures, can be built. A major computer science issue is whether such computers can be used effectively to enhance the performance of knowledge-based systems. Since 1985, the Knowledge Systems Laboratory at Stanford University has been investigating these issues.

The goals and technical approach of this project, largely supported by DARPA under the Strategic Computing Program, have been to achieve two to three orders of magnitude speed-up in the execution of knowledge-based systems, by identifying and exploiting sources of concurrency at all levels of system design: the application level, the problem-solving framework level, the programming language level and the hardware systems architecture level. Due to the inherent complexity of the task and the lack of theoretical foundations for parallel computation with ill-structured problems, we have taken an empirical approach.

Simulation of systems at an architectural level offers an effective way to study critical design choices. SIMPLE/CARE is a powerful simulation system that forms the foundation for our empirical investigations. SIMPLE is a CAD (Computer Aided Design) system for hierarchical, multiple-level specification of computer architectures and includes an associated mixed-mode, event-based simulator. CARE is a parameterized, multiprocessor array emulation defined in SIMPLE's specification languages and running on SIMPLE's simulator. Our simulation system has been used internally to make quantitative comparisons of the performance of various architectures and to gain insights into how different concurrent programming models support the development of concurrent applications decomposition and organization. The system is in use by several research groups at Stanford and it has been exported to several other sites, including NASA Ames Research Center. A tutorial was held in January 1988, attended by representatives from the DoD, NASA and Boeing, which described the CARE/SIMPLE system. The attendees received instruction in use of the system for making measurements of the performance of various simulated multiprocessor applications. A Stanford graduate course on these tools is currently in progress this spring quarter.

III.B.4. Toward the Distributed SUMEX-AIM Community

We have made a key decision this past year on the core system definition that will support the first phase of the distributed AIM community. Guided by our requirements to provide powerful and widely-available tools for general computing and biomedical research and to sharply focus our limited development resources on a small number of standardized hardware and software configurations, we considered a wide range of alternative systems for AIM community computing needs to replace and upgrade the services of the 2060. Based on dominant user preferences for the icon-based interface, outstanding technical performance, very competitive academic pricing, and an already-growing group of national AIM users, we have chosen Apple Macintosh II workstations as the general computing environment for researchers and staff, TI Explorer Lisp machines (including the microExplorer Macintosh coprocessor) as the near-term high-performance Lisp research environment, and a SUN-4 as the central system network server (network services, file services, printing services, etc.). To actually implement a prototype of the planned distributed environment, a substantial quantity of hardware was purchased with DARPA research funds in the spring of 1988. We are now in the midst of the installation and integration process, concentrating initially on getting basic capabilities operational, such as for text processing, filing/archiving, printing, graphics, office management, system building tools, information resource access, and distributed system operation and management.

Initial user response to the introduction of these systems has been overwhelmingly enthusiastic, even though there are many "rough edges" remaining to be smoothed out in the systems integration. Our core development work for the environment of the Mac II-, Explorer-, and SUN-based system has focussed on providing remote access between workstations themselves and with servers, integrating a solid support of the TCP-IP network protocol, and building a powerful distributed electronic mail system. The new Mac II mail system will be an adaptation of the prototype distributed system developed in recent years for the Xerox Lisp machine and which is in routine use by a number of people and is being ported to the Explorer.

One of the key issues in selecting the systems for our distributed computing environment was the performance of Common Lisp and to help make this evaluation, we undertook an informal survey of the performance of two KSL AI software packages, SOAR and BB1, on a wide variety of machines. Within a factor of two of the best performance, a considerable range of workstations based on stock microprocessor chips (e.g., the Motorola 680xx and the Intel 80386) as well as specially microprogrammed Lisp chips have comparable performance. Even though performance gaps between microprogrammed Lisp systems and stock workstation implementations are narrowing, there still remains a significant difference in the quality of the development environments. We have attempted to distill and promote the commercial development of the key features of the Lisp machine environments that would be needed in stock machine implementations in order to make them attractive in a development setting.

After the prototype distributed system is implemented and tested in the Stanford KSL environment, we will package and document its elements so that other sites in the AIM community and beyond can duplicate its capabilities. As this work progresses we will phase out the old DEC 2060 to be replaced by the SUN-4 as a general community communication and information server.

III.C. Administrative Changes

There have been few administrative changes within the project this past reporting year. Professor Shortliffe had been on sabbatical at the University of Pennsylvania the year before last and returned to Stanford in mid-July 1987, when he resumed his role as SUMEX Principal Investigator.

We continue to operate the cost recovery system we reported on last year as part of phasing out BRTP subsidy of the DEC 2060 facility. The details of this system are discussed on page 121. In summary, we are successfully recovering the projected 40% of 2060 operations costs this year ($136,374) from Stanford users, with the declining component of NIH support (60% this year) used to protect national users from fees for service, including communications. This additional burden on Stanford projects continues to be absorbed almost entirely in existing direct cost budgets since no supplements for new computing costs were forthcoming in the middle of on- going grant and contract awards. This has affected staffing and student support directly in our labor-intensive research efforts. All of our new support applications are being written with requests for funds to cover projected computing charges.

This next year we will increase the cost recovery goal to 60% of projected 2060 operations costs as scheduled in our grant application of June 1985. We also plan to physically phase out the 2060 and replace it with the new SUN-4 network server if technical development activities follow on schedule. The detailed interaction of this transition with our cost recovery procedures remain to be worked out during this coming year.

III.D. Resource Management and Allocation

III.D.1. Overall Management Plan

Early in the design of the SUMEX-AIM resource, an effective management plan was worked out with the Biotechnology Resources Program (now Biomedical Research Technology Program) at NIH to assure fair administration of the resource for both Stanford and national users and to provide a framework for recruitment and development of a scientifically meritorious community of application projects. This structure has been described in some detail in earlier reports and is documented in our recent renewal application. It has continued to function effectively as summarized below.

- The AIM Executive Committee meets periodically by teleconference to advise on new user applications, discuss resource management policies, plan workshop activities, and conduct other community business. The Advisory Group meets as needed to review project applications. (See Appendix D for a current listing of AIM committee membership).

- We actively recruit new application projects and disseminate information about AI in biomedicine. With the development of more decentralized computing resources within the AIM community outside of Stanford, the use of SUMEX resources by AIM members has shifted more and more toward communication with colleagues and access to information.

- With the advice of the Executive Committee, we have opened SUMEX-AIM resources widely to biomedical users desiring electronic communications facilities. A list of current users who have used SUMEX for this purpose over this past year is given starting on Page IV.E.

- We have carefully reviewed on-going projects with our management committees to maintain a high scientific quality and relevance to our biomedical AI goals.

- We continue to provide active support for the AIM workshops. The most recent one was held this spring at Stanford University, under the auspices of the American Association for Artificial Intelligence (AAAI).

- We have continued to provide systems advice to users attempting to set up computing resources at their own sites, based on the expertise developed in the SUMEX resource environment.

- We have tailored resource policies to aid users whenever possible within our research mandate and available facilities.

III.D.2. 2060 Cost Center

General Cost Center Structure

Our plan for the term of the current grant is a firm but responsible transition of the SUMEX-AIM resource to a distributed community model of operation. There has continued to be a group of national and local users -- particularly young projects needing seed support prior to obtaining major funding -- that depend on a central shared resource like the SUMEX mainframe. In addition, the 2060 has played a key role as a central server for intercommunity communication and shared information. Powerful and widely available workstation equipment is rapidly becoming accessible, however, at a cost that most projects can afford, even young ones. Thus, the period of critical dependence on the DEC 2060 for raw computing cycles is largely past and its role in supporting routine computing and communication services can also soon be replaced by other more cost effective means. We are in the process of phasing out the SUMEX 2060 machine over the next few years in favor of the new distributed workstation environment we are developing. This process is progressing gradually and responsibly so that our users can relocate to other facilities or move to workstation environments for their research without disruption.

Specifically, our renewal proposal for the five-year period 8/1/86-7/31/91, submitted to the Division of Research Resources in June 1985, called for phasing out NIH support for DEC 2060 mainframe operations over the course of the grant period and the establishment of a cost center at Stanford to recover the unsubsidized costs of 2060 operations from the established Stanford user community. This phase-out process is taking place linearly over five years, with 80% of the 2060 costs charged to the resource budget in renewal year 1 (Grant Year 14), 60% in year 2 (current grant year), 40% in year 3, 20% in year 4, and 0% in year 5, when routine operations (even national user services) will be supported entirely by user revenues. Use of the 2060 by members of the national AIM community is still free of charge at this time, and we will continue to cover the total cost of national community 2060 usage from the NIH subsidy as long as funding permits.

In keeping with this plan, during the summer of 1986, we requested and received the approval of the Government Cost and Rate Studies section of Stanford's Controller's Office to establish a 2060 cost center effective at the start of renewal year 1 (August 1, 1986) with a charge per CPU hour based on our projections of 2060 operations costs and anticipated billable CPU usage.

In last year's annual report, we reported success in recovering the 20% of 2060 operations costs not subsidized by NIH from our Stanford user community during our first year of cost center operation. This year our objective was to recover 40% of 2060 costs from Stanford users with a corresponding increase in the rate charged per CPU hour as of August 1, 1987, the start of renewal year 2. We have been monitoring cost center expenses and revenues carefully this second year and again anticipate breaking even at the end of the cost center's fiscal year at the end of July. Figure 19 shows the cumulative user revenues collected by month for the period August 1987 through April 1988.

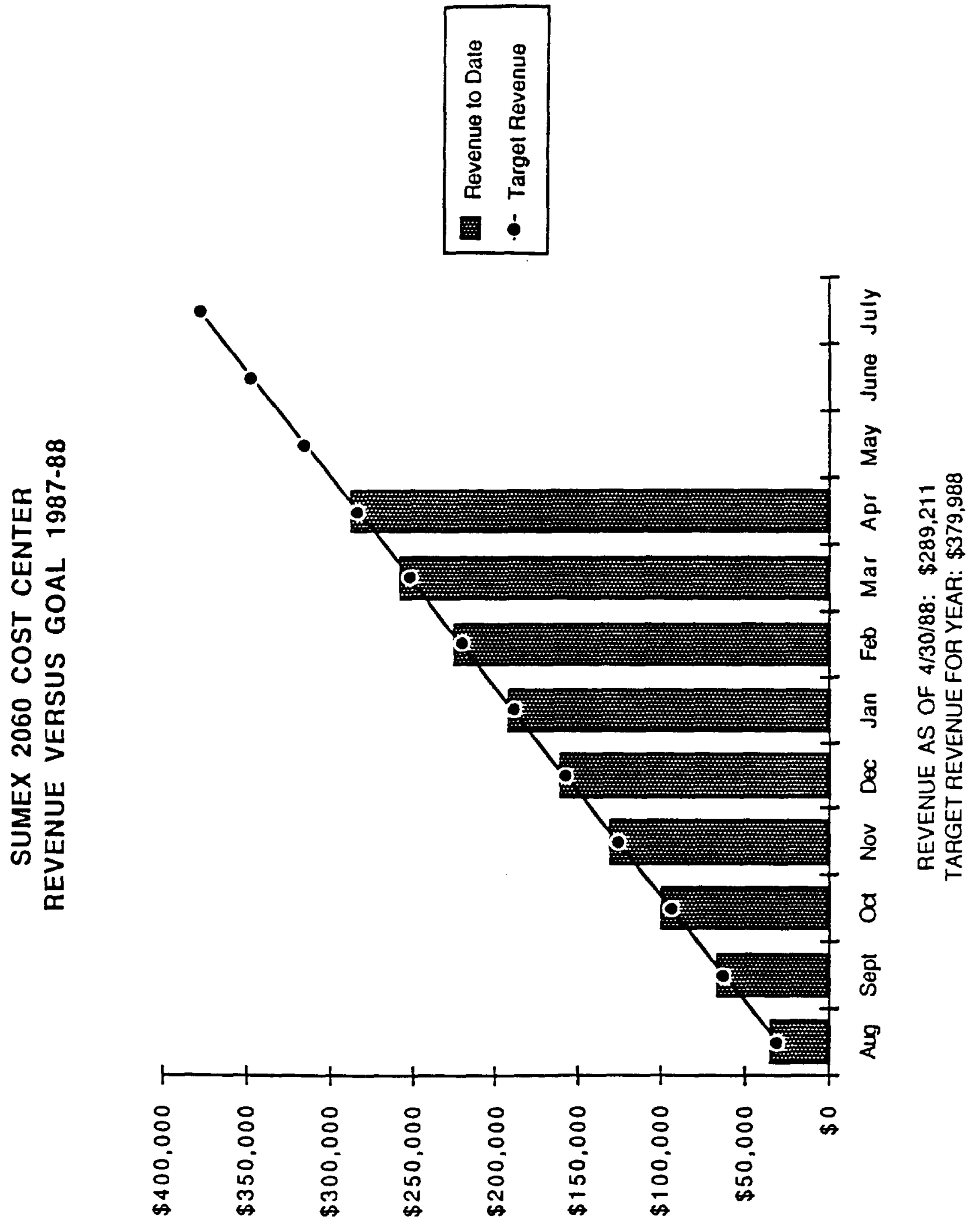

Figure 19: 2060 Cost Center Performance

III.E. Dissemination of Resource Information

We are continuing our past practice of making a substantial effort to disseminate the AI technology developed here. This has taken the form of many publications -- over forty-five combined books and papers are published per year by the KSL; wide distribution of our software, including systems software and AI application and tool software, both to other research laboratories and for commercial development; production of films and video tapes depicting aspects of our work; and significant project efforts at studying the dissemination of individual applications systems such as the ONCOCIN resource-related research project (see 144).

Software Distribution

We have widely distributed both our system software and our AI tool software. Since much of our general system-level software is distributed via the ARPANET we do not have complete records of the extent of the distribution. Software such as TOPS-20 monitor enhancements, the Ethernet gateway and TIP programs, the SEAGATE AppleNet to Ethernet gateway, the PUP Leaf server, the SUMACC development system for Macintosh workstations, and our Lisp workstation programs are frequently distributed in this manner to the ARPANET community and beyond.

Our primary distribution effort is directed towards the AI tools we have developed. In recent years, the volume of inquiries for this type of software and requests for tapes has been a substantial burden on the staff and so it was decided to turn over most of this type of software distribution to Stanford's Office of Technology Licensing (OTL). This organization handles software distribution and technology licensing matters for much of the Stanford community. Since there are several OTL staff members assigned to the distribution of Stanford software, requests for information and tapes are handled quickly and efficiently. Also, OTL's staff has the expertise needed to handle the legal questions that frequently arise in the distribution of software, and an established computerized record-keeping scheme. SUMEX staff continues to be available as needed to assist OTL with special administrative and technical matters.

Specific software distribution events this past year include:

- The Parallel Computing Architectures Project multiprocessor simulation system, CARE/SIMPLE, is in use by several research groups at Stanford and it has been ported to several external sites including NASA Ames Research Center.

- Two (2) licenses were granted for the EMYCIN package and twenty-two (22) licenses were granted for the BB1 package.

- OTL has concluded a license arrangement with Cisco, Inc. for the commercial development and marketing of the SUMEX Ethernet gateway and TIP service software.

- The agreement between Stanford University and Kinetics Inc. covering hardware and software technology for an Ethernet-to-AppleTalk gateway has been converted from an on-going royalty agreement to a fully paid license.

- OTL reported the expiration of an exclusive license to Molecular Designs Ltd. covering some aspects of the software generated by the DENDRAL Project. The source code and binary versions of this software are now available to all users (commercial, government, and academic) through OTL.

AIM Community Systems Support

We continue to make a special effort to assist other members of the SUMEX-AIM community in integrating the technologies needed for biomedical AI research. This is often achieved through direct contact with staff members at these institutions, (e.g., with Professor Sticklen's group at Michigan State University after his move from Ohio State and with Professor Widman's group at the University of Texas), at meetings and workshops, or via electronic mailing lists. For example, the Info-MAC, Info-Explorer, and Info-1100 mailing lists have hundreds of members and cover a broad range of equipment issues, software issues, and topics in artificial intelligence.

Video Tapes and Films

The KSL has continued to prepare video tapes that provide an overview of the research and research methodologies underlying our work and that demonstrate the capabilities of particular systems. These tapes are available through our groups, the Fleischmann Learning Center at the Stanford Medical Center, and the Stanford Computer Forum, and copies have been mailed to program offices of our various funding sponsors. In addition to the earlier tapes covering Knowledge Engineering in the KSL, ONCOCIN Overview, and ONCOCIN Demonstration, we have recent tapes on the PROTEAN project, the BB1 project, and a one-day symposium on KSL research activities.

III.F. Suggestions and Comments

Resource Organization

We continue to believe that the Biomedical Research Technology Program is one of the most effective vehicles for developing and disseminating technological tools for biomedical research. The goals and methods of the program are well-designed to encourage building of the necessary multi-disciplinary groups and merging of the appropriate technological and medical disciplines.

Electronic Communications

SUMEX-AIM has pioneered in developing more effective methods for facilitating scientific communication. Whereas face-to-face contacts continue to play a key role, in the longer-term computer-based communications will become increasingly important to the NIH and the distributed resources of the biomedical community. We would like to see the BRTP take a more active role in promoting these tools within the NIH and its grantee community. This is particularly important in the light of significant on-going changes to the national networking environment (see Page 75).